



TAMPEREEN  
AMMATTIKORKEAKOULU

# **PALKANMAKSUOHJELMAN TESTAUS JA TUOTEKEHITYS**

Taru Kivinen

Opinnäytetyö  
Toukokuu 2016  
Tietotekniikan koulutusohjelma  
Tietoliikennetekniikka



## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietotekniikan koulutusohjelma  
Tietoliikennetekniikka

KIVINEN TARU:

Palkanmaksuohjelman testaus ja tuotekehitys

Opinnäytetyö 23 sivua, joista liitteitä 0 sivua  
Toukokuu 2016

---

Opinnäytetyössä selvitettiin palkanmaksuohjelman testaus ja kehitys joukkoliikennöintiyritykselle yhteistyössä Tampereen ammattikorkeakoulun ohjelmistotekniikan opiskelijoiden kanssa. Tavoitteena oli tuottaa alan määräyksien mukainen ohjelma, joka tuottaa valmiit palkkalaskelmat ohjelmaan syötettyjen työvuorolistojen perusteella. Ohjelman tekoon käytettiin Meteor-selainkehitysympäristöä, Node.js-ohjelmointiympäristöä sekä MongoDB-tietokantaa. Ohjelman teossa ei huomioitu tarpeeksi hyvin kokonaistiedon määrää, jonka vuoksi ohjelman valmistumisen ja käyttöönoton aikataulu viivästyi. Lisäksi ohjelmassa oli rakenteellisia ongelmia, joiden korjaamiseen kului aikaa.

Ohjelmaan syötettävää dataa ovat työntekijöiden tiedot sekä liikennöivät linjavuorot. Näiden tietojen perusteella voidaan luoda työvuorolistat, ja suorittaa palkanmaksulaskelmat kullekin työntekijälle. Ohjelma on tehty mahdollisimman helppokäyttöiseksi ja sen käyttöönotto sujuvaksi antamalla lisätä tietoja myös jälkikäteen tai erikseen esimerkiksi kertyneiden vapaapäivien muodossa.

Ilmenneiden ongelmien vuoksi, ohjelma ei opinnäytetyön palauttamisen ajankohtana ole vielä loppuasiakkaan käytössä, mutta todennäköisesti ohjelma valmistuu muutaman kuukauden kuluessa. Ohjelmaa voidaan kehittää jatkossa myös muille yrityksille ottaen huomioon asiakkaiden yksilölliset tarpeet.

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Degree Programme in ICT engineering  
Telecommunications and networks

**KIVINEN TARU:**

Research and development of a salary payment program.

Bachelor's thesis 23 pages, appendices 0 pages  
May 2016

---

The purpose of the thesis was to test and develop a salary payment program for a bus company/public transport service company. The program is developed in collaboration with software engineering students of Tampere University of Applied Sciences. The aim was to produce field regulations according program, which produces salary slips on the basis of rosters added to the program. Used softwares for making the program were Meteor web framework, runtime environment Node.js and database MongoDB. While making the program, the full amount of information was not taken into account, which due to delay of the schedule. In addition the program had some structural disabilities, which took time to repair.

The data needed by the program were the employee information and operated routes. According to these, the rosters can be made and the salary slips can be calculated for each employee. The program is made as easy as possible to use and new information details can be added to it afterwards or separately, when, for an example, the program has been taken in use and some employees already have unused paid days off.

Because of the problems that encountered, the program is not available to the customer when this thesis has been released. It is however likely, that the program will be ready within the next two months. In the future the program can be developed for other companies as well, considering the individual needs of the customers.

---

Key words: salary payment software, software engineering

## SISÄLLYS

1	JOHDANTO.....	5
2	OHJELMAN SISÄLTÖ .....	6
2.1	Ohjelman syöte .....	6
2.2	Huomioon otettavia asioita .....	6
2.2.1	Palkka.....	6
2.2.2	Työaika.....	7
2.2.3	Vapaan kertyminen .....	7
3	OHJELMAN TOTEUTUS .....	10
3.1	Node.js-ohjelmointiympäristö .....	10
3.2	MongoDb-tietokanta .....	11
3.3	Meteor-selainkehitysympäristö.....	12
4	OHJELMAN TOIMINTA.....	13
4.1	Linjavuorot.....	13
4.2	Työntekijät .....	15
4.3	Työvuorolistat .....	18
4.4	Palkanmaksu .....	19
5	TUOTEKEHITYS .....	20
5.1	Käyttäjäystävällisyys .....	20
5.2	Muille yrityksille.....	20
5.3	Ylläpito ja muutokset.....	21
6	POHDINTA.....	22
	LÄHTEET.....	23

## 1 JOHDANTO

Tässä opinnäytetyössä on toteutettu palkanmaksuohjelma joukkoliikennöintiyritykselle. Sovellus on tehty yhteistyössä Tampereen ammattikorkeakoulun ohjelmistotekniikan opiskelijoiden kanssa. Opinnäytetyössä käsitellään ohjelman rakennetta, toteutusta ja sen jatkokehitysmahdollisuuksia.

Palkanmaksuohjelman keskeisimpänä vaatimuksena oli tuottaa palkkalaskelmia ohjelmaan syötettyjen työvuorolistojen perusteella. Ohjelman tuli toimia työehtosopimuksen, lainsäädännöllisten, yrityksen omien käytäntöjen ja alalle ominaisten määräysten mukaisesti.

Ohjelma toteutettiin valmiiksi yhteensopivilla ohjelmistoilla, joihin kuului selainkehitysympäristö Meteor, ohjelmointiympäristö Node.js sekä tietokanta MongoDB. Ohjelmointikielenä on näille ohjelmille ominainen JavaScript-kieli. Palvelimella toimii myös Nginx-välityspalvelin.

## **2 OHJELMAN SISÄLTÖ**

### **2.1 Ohjelman syöte**

Palkanmaksuohjelma on tehty liikennöintiyritykselle, ja ohjelman kaksi olennaisinta käyttäjän lisäämää tietokantaa ovat yrityksen liikennöimät linjavuorot sekä työntekijätiedot. Näillä tiedoilla ohjelmassa voidaan luoda työvuorolistat, joihin palkanmaksu pohjautuu.

Palkanmaksuohjelma ei suorita varsinaista palkanmaksua, vaan laskee palkat valmiiksi palkanmaksusta vastaavalle henkilölle.

### **2.2 Huomioon otettavia asioita**

Ohjelmaa kirjoitettaessa on otettu huomioon linja-autohenkilökunnan työehtosopimus, joka määrittelee muun muassa vähimmäispalkkauksen, vähimmäis- ja enimmäistyöajan, pisimmän yhtäjaksoisen työajan sekä lepoajan. Lisäksi työn tilaajan omat käytännöt on otettu ohjelman teossa huomioon.

#### **2.2.1 Palkka**

Työehtosopimus määrittelee useita palkkaukseen liittyviä asioita, jotka on otettava huomioon palkanmaksuohjelmaa tehtäessä. Palkka määräytyy työtoimenkuvan ja työuran pituuden mukaisesti. Työntekijän palkka nousee, mitä pidempään hän on ollut töissä. Linja-autohenkilökunnan tapauksessa työuran kesto jaetaan yleensä neljään eri palkkaluokkaan, jotka ovat alle 4 vuotta, 4-8 vuotta, 8-12 vuotta ja yli 12 vuotta.

Työehtosopimuksessa määritellään myös vähimmäiskorvaukset ilta-, yö-, sunnuntai- sekä ylityöstä, vapaapäivänä tehdystä työstä maksettava vapaapäivätyölisä sekä joulutyölisä.

Ohjelma on tehty Tampereen lähiseudulla toimivalle liikennöintiyritykselle, ja siihen voidaan soveltaa myös työehtosopimuksen määrittelemää kohtaa Tampereen yhteistariffiliikenteestä, eli ajalta, jonka Tampereen yhteistariffiliikenteessä työskentelevä kuljettaja ajaa ko. yhteistariffialueella, suoritetaan 0,51 euron suuruinen lisä tuntia kohden.

(AKT 2014, 7)

Sairaslomapäivien ajalta maksettavaa palkkaa ei ole huomioitu ohjelman toiminnassa, vaan siitä vastaa palkanmaksaja alan säädöksiin kuuluvalla tavalla.

Palkkalaskelma ilmoittaa palkasta tehtävät vähennykset, joita ovat liiton jäsenmaksut, eläkemaksut, työttömyysvakuutusmaksu sekä ennakonpidätys.

### **2.2.2 Työaika**

Työehtosopimus määrittelee enimmäistyöajan yhtämittäiselle työlle ja peräkkäisille työpäiville, sekä vähimmäislepoajan työvuorojen välissä. Nämä on otettu ohjelman teossa huomioon, eikä ohjelma anna käyttäjän lisätä työntekijälle työehtosopimuksen vastaisia työvuoroja tai työvuoroputkia. Yleisesti työehtosopimus määrittelee enimmäistyöajaksi 11 h, sekä yhtämittaisen työn enimmäispituudeksi 5,5 h. (AKT 2014, 8.)

### **2.2.3 Vapaan kertyminen**

Työntekijän vapaapäivien kertymiseen on työehtosopimuksessa kaksi kuvien 1 ja 2 mukaista taulukkoa, joissa määritellään kertymä sen mukaan, onko kyseisenä vuonna loppiainen joko lauantaina tai sunnuntaina (kuva 1) tai muuna arkipäivänä (kuva 2).

## A. Loppiainen lauantaina tai sunnuntaina

	työtuntimäärä		lyhennys			
vähintään	94	h	8	h	eli	1 päivä
	188	h	16	h	eli	2 päivää
	283	h	24	h	eli	3 päivää
	377	h	32	h	eli	4 päivää
	471	h	40	h	eli	5 päivää
	565	h	48	h	eli	6 päivää
	660	h	56	h	eli	7 päivää
	754	h	64	h	eli	8 päivää
	848	h	72	h	eli	9 päivää
	942	h	80	h	eli	10 päivää
	1036	h	88	h	eli	11 päivää
	1131	h	96	h	eli	12 päivää
	1225	h	104	h	eli	13 päivää
	1319	h	112	h	eli	14 päivää
	1413	h	120	h	eli	15 päivää
	1508	h	128	h	eli	16 päivää
	1602	h	136	h	eli	17 päivää
	1696	h	144	h	eli	18 päivää

KUVA 1. Vapaapäivien kertymä, mikäli kyseisenä vuonna loppiainen on lauantaina tai sunnuntaina (AKT 2014, 11–12).



## B. Loppiainen muuna arkipäivänä kuin lauantaina

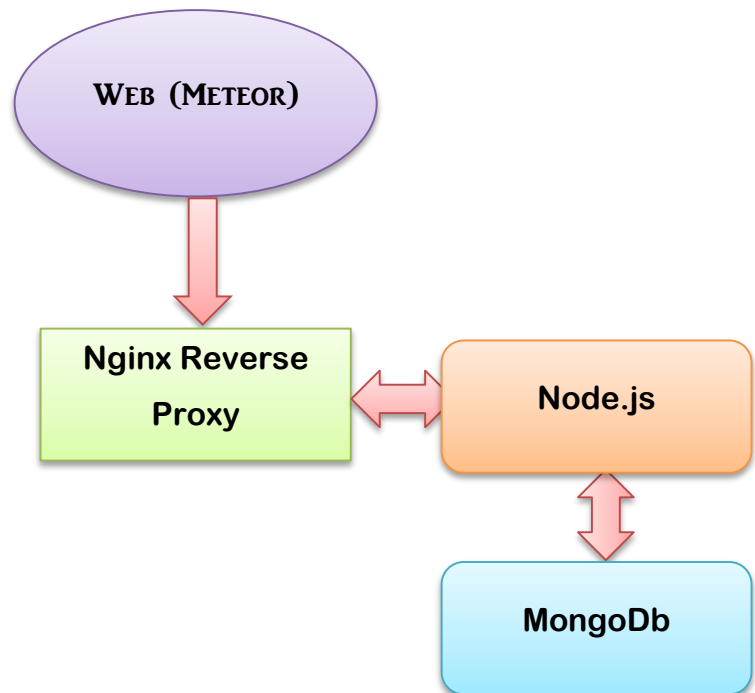
	työtuntimäärä		lyhennys			
vähintään	90	h	8	h	eli	1 päivä
	180	h	16	h	eli	2 päivää
	270	h	24	h	eli	3 päivää
	360	h	32	h	eli	4 päivää
	447	h	40	h	eli	5 päivää
	537	h	48	h	eli	6 päivää
	627	h	56	h	eli	7 päivää
	717	h	64	h	eli	8 päivää
	807	h	72	h	eli	9 päivää
	893	h	80	h	eli	10 päivää
	983	h	88	h	eli	11 päivää
	1073	h	96	h	eli	12 päivää
	1163	h	104	h	eli	13 päivää
	1253	h	112	h	eli	14 päivää
	1340	h	120	h	eli	15 päivää
	1430	h	128	h	eli	16 päivää
	1520	h	136	h	eli	17 päivää
	1610	h	144	h	eli	18 päivää
	1696	h	152	h	eli	19 päivää

KUVA 2. Vapaapäivien kertymä, mikäli kyseisenä vuonna loppiainen on muuna arkipäivänä kuin lauantaina (AKT 2014, 11–12).

Nämä tiedot on lisättävä ohjelmaan erillisinä ehto-rakenteinaan, koska taulukot eivät ole lineaarisia tai matemaattisesti merkittävissä.

### 3 OHJELMAN TOTEUTUS

Ohjelma on toteutettu avoimen lähdekoodin Node.js-ohjelmointialustalla, ja tietokannat on toteutettu MongoDB-ohjelman avulla. Työskentelyalustana on käytetty Meteor-ohjelmaa ja palvelimella toimii Nginx reverse proxy -välityspalvelin, joka ohjaa pyynnöt Node.js-palvelimelle (kuva 3).



KUVA 3. Ohjelmiston hierarkia.

#### 3.1 Node.js-ohjelmointiympäristö

Node.js on alustariippumaton ohjelmointiympäristö palvelinpuolen Internet-sovelluksien luomiseen. Useimmat sen perusmoduuleista on kirjoitettu JavaScript-ohjelmointikielellä, jonka tulkintaan käytetään Google V8 JavaScript -konetta. Node.js voidaan yhdistää selaimen ja tietokannan, kuten MongoDB, kanssa toimivaksi kokonaisuudeksi. (Wikipedia 2016a.)

Node.js käsittelee työskentelyalusta Meteorilla toteutetut JavaScript-ohjelmointikielen ohjelmointikoodit.

### 3.2 MongoDB-tietokanta

MongoDb on avoimella lähdekoodilla toteutettu alustariippumaton ja dokumenttikeskeinen tietokanta. Sen rakenne ei ole perinteinen taulukkopohjainen relaatiotietokanta, vaan skaalautuva ja joustava dynaaminen kaaviomalli. (Wikipedia 2016b.)

Dokumentit, eli syötteet, voivat sisältää yhden tai useita tietokenttiä, kuten jonoja, binäärimuotoista dataa ja alidokumentteja, jotka voivat vaihdella eri dokumenttien välillä. Tämä mahdollistaa tietokannan paremman kehityksen. (MongoDb 2016.)

MongoDb on kehitetty käsittelemään kasvavia tietomääriä ja toimimaan hyvällä suorituskyvyllä. Ohjelman tavoite on olla mahdollisimman helppokäyttöinen. Relaatiotietokannoissa tieto tallennetaan taulukkoihin, kun taas MongoDB:ssä dokumentit tallennetaan kokoelmiin (kuva 4). (Ari Hovi 2016.)

TIETOKANTOJEN EROT		
<b>Relaatiokanta</b>		
<b>Asiakkaat</b>		
..	..	..
2799	Oy Firma Ab	..
..	..	..
<b>Tilaukset</b>		
..	..	..
13576	..	2199
..	..	..
<b>Maksutavat</b>		
13576	Visa	447361198
..	..	..
<b>Dokumenttikanta</b>		
<b>Tunnus:</b>	13756	
<b>Asiakas:</b>	Oy Firma Ab	
<b>Tilaukset</b>		
10071	2	57 €
21198	1	41 €
10922	2	41 €
<b>Maksutavat</b>		
<b>Kortti:</b>	Visa	
<b>Nro:</b>	447361198	
<b>RELAATIO VAIN DOKUMENTTI.</b> Relaatiokannassa eri asiat talletetaan eri tauluihin ja yhdistetään hakiessa. Dokumenttikannassa yhteenliittyvät asiat voi tallettaa samaan dokumenttiin.		

KUVA 4. Tietokantojen erot (Ari Hovi 2016).

### **3.3 Meteor-selainkehitysympäristö**

Meteor on alustariippumaton avoimen lähdekoodin selainkehitysympäristö, jota voidaan ohjelmoida Node.js -ohjelman kautta Javascriptillä. Se on valmiiksi yhteensopiva myös MongoDB-tietokantapalvelimen kanssa. (Wikipedia 2016c.)

## 4 OHJELMAN TOIMINTA

### 4.1 Linjavuorot

Yritys voi lisätä ohjelmaan liikennöimänsä reitit itse. Toiminto on tehty erittäin helpoksi sivustonäkymän Reitit-välilehdessä.

Lisättäessä uutta reittiä kuvassa 5, tulee antaa tiedot reitin pituudesta, kestosta sekä ajan- kohdasta.

The screenshot shows a web browser window with the application running on localhost:3000/routes/insert. The page title is 'Lisää uusi reitti'. The form contains the following fields and options:

- Tunnus: 321
- Alkuaika: 10:12
- Loppuaika: 21:00
- Perustunnit: 10
- Sunnuntai tunnit: 0
- Iltaisiä: 1
- Yöisiä: 2
- Odotus tunnit: 2
- Päiväraha: 100
- Kilometrit: 1.2
- Tamperelisiä: 1.2
- Yityöisiä: ☐
- Maanantai: ☒ Tilistai: ☒ Keskiyö: ☒ Torstai: ☒ Perjantai: ☒ Lauantai: ☐ Sunnuntai: ☐
- Tarkistettava listalla: ☒
- Buttons: Tallenna (green), Peruuta (grey)

KUVA 5. Kuvakaappaus Reitit-välilehden ”Lisää uusi”-kohdasta.

Lisätessä uutta reittiä (kuva 6), tulee antaa reitin tai linjavuoron yksilöllinen numero, sen liikennöinnin aloitusaika, lopetusaika ja kokonaiskesto. Vaikka samaa reittiä kuljet- taisiin sekä yö- että päiväsaikaan, tai pyhäpäivisin, tulee kaikki nämä ajoajankohdat il- moittaa erillisinä reitteinä ohjelmalle, mikäli reitiltä maksetaan eri lisiä eri aikoina. Käy- tännössä siis esimerkiksi sunnuntaina ajettava vuoro annetaan ohjelmalle erillisenä linja- vuoronaan, vaikka ohjelmalla olisi jo tiedossa sama reitti arkipäiväajolle.

Työaika & vuorosarjat
Koti
Työntekijät
Kalusto
Reitit
Vuorolistat
Palkanmaksu
Raportit

## Lisää uusi reitti

Tunnus

123

Alkuaika:

08:00

Loppuaika:

16:00

Perustunnit

8

Sunnuntai tunnit

0

Iltalisä

0

Yölisä

0

KUVA 6. Kuvakaappaus reitin lisäys -sivun yläosasta.

Mikäli reitillä ei voida jatkaa matkaa välittömästi pysäkkien välillä, merkitään ohjelmaan kuljettajan odotustunnit. Esimerkkikuvassa 7 on merkitty odotustunneiksi 2 h, eli esimerkiksi kuljettaja joutuu odottamaan Valkeakoskelta Tampereelle liikennöidessään Tampereella 2 h ennen paluumatkan alkamista. Lisäksi ohjelmaan annetaan reitiltä kertyneet kilometrit, ja tieto ”Tampere-lisästä” (kohta 2.2). Erikseen on valittavissa myös, mikäli reitille kuuluu tai on kuulunut ylityötä, ja minä päivinä reittiä liikennöidään. Valittaessa sunnuntai-päivän, tulee muistaa merkitä kuvan 6 kohtaan ”sunnuntailisä” kertyneet sunnuntaitunnit. Myös mahdollisten päivärahojen maksu työntekijälle huomioidaan linjavuoro-osiossa.

**Odotus tunnit**

**Päiväraha**

**Kilometrit**

**Tampere-lisä**

☐ Ylityölisä

☒ Maanantai ☐ Tiistai ☒ Keskiviikko ☐ Torstai ☐ Perjantai ☒ Lauantai ☐ Sunnuntai

☒ Tarkistettava listalla

TKI-Verstas 2015

Suosittellemme Google Chrome Selainta

KUVA 7. Kuvakaappaus reitin lisäys -sivun alaosasta.

Kun liikennöidyt reitit on saatu lisättyä ohjelmaan, on Reitit-välilehden näkymä kuvan 8 kaltainen.

Työaika & vuorosarjat										
Koti Työntekijät Kalusto Reitit Vuorolistat Palkanmaksu Raportit Asetukset										
+ Lisää uusi		Haku								
Tunnus	Perustunnus	Sunnuntai tunnit	Ilta-lisä	Yö-lisä	Odotus tunnit	Kilometrit	Tampere-lisä	Vuorotyö-lisä	Tarkistettava	
123	8	0	0	0	2	100	1,03	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="✎"/> <input type="button" value="🗑"/>
321	10	0	1	2	2	100	1,2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="✎"/> <input type="button" value="🗑"/>

KUVA 8. Kuvakaappaus reitit-välilehdestä.

## 4.2 Työntekijät

Työntekijä voidaan lisätä ohjelmaan Työntekijät-välilehdeltä samaan tapaan kuin reitin lisäys. Kuvassa 9 on näkymä työntekijän lisäys -toiminnon aloituksesta.

Työaika & vuorosarjat	Koti	<b>Työntekijät</b>	Kalusto	Reitit	Vuorolistat	Palkanmaksu	Raportit
-----------------------	------	--------------------	---------	--------	-------------	-------------	----------

Nimi

Tero Testaaja

Tunnus

TTES

Osoite

Testikatu 123

SOTU

123123-123

Aloituspäivä

11/11/2011

Verotuskunta

Tampere

BIC-numero

BIC

Iban-tilinumero

FI00 0000 0000 0000 00

Tuloraja

2345€

KUVA 9. Kuvakaappaus sivun näkymästä, kun lisätään uutta työntekijää.

Ohjelmaan syötetään henkilötiedot ja mahdollinen tunniste, työsuhteen alkamispäivä, verotuskunta sekä pankkitilin tiedot BIC- ja IBAN-sarjoilla ilmoitettuna. Myös verokortin tuloraja tulee ilmoittaa. Kuvassa 10 on sivuston näkymän alaosa.



**Verotuskausi**

Palkkakausikohtainen

**Perusveroprosentti**

20%

**Lisäveroprosentti**

60%

**Palkanmaksujakso (vk)**

2vk

**Kertyneet pekkaspäivät**

**Keskituntiansio**

12.3€

**Palkkaryhmä**

**Liitto**

☐ Kuljettaja

☐ Korjaamo henkilökunta

☐ Toimihenkilö

☐ Aktiivinen

☐ Ulosotto

KUVA 10. Kuvakaappaus sivun näkymästä, kun lisätään uutta työntekijää.

Työntekijän palkanmaksutietoihin tulee ilmoittaa verotuskausi, veroprosentit, palkanmaksujakson pituus ja keskituntiansio. Ohjelma laskee itse työntekijän kertyneet vapaapäivät, mutta mikäli työntekijälle on kertynyt pekkaspäiviä ennen ohjelman käyttöönottoa, tulee ne merkitä manuaalisesti työntekijän omiin tietoihin.

Ohjelman tilanneella yrityksellä on yrityksessään kolme eri palkkaryhmää työntekijöiden toimikuvan mukaan. Näitä ovat kuljettaja, korjaamon henkilökunta sekä yleinen toimihenkilö.

Kuvassa 10 nähdään teksti ”Liitto”, jossa pitäisi myös olla valintamahdollisuus. Palkanmaksuohjelma laskee alan omaan työntekijäliittoon kuuluvien työntekijöiden liittomaksun osuuden palkasta.

Valintaruudun ”aktiivinen” rastittamalla työntekijä tulee näkyviin ohjelman muissa osissa, kuten työvuoronsuunnittelussa. Mikäli työntekijä ei jostakin syystä ole esimerkiksi pidemmällä aikavälillä käytettävissä, voi rastin ottaa ruudusta väliaikaisesti pois.

”Ulosotto”-valinta voidaan valita, mikäli työntekijällä menee tietty osuus ansioista ulosottopidätykseen. Palkanmaksuohjelma ei laske ulosoton määrää, vaan antaa ilmoituksen pidätyksestä palkanmaksajalle tai palkanmaksutoimintoa käyttävälle.

### 4.3 Työvuorolistat

Työvuorot-välilehdessä ohjelman käyttäjä voi luoda työvuorolistoja ohjelmaan syötettyjen työntekijöiden ja linjavuorotietojen perusteella.

Osiossa voidaan luoda uusi työvuorolista tai muokata aiemmin luotua listaa. Listat tehdään aina kahdeksi viikoksi kerrallaan. Lisättäessä uutta työvuorolistaa, sivustonäkymän vaaka-akselilla on lueteltu kahden viikon viikonpäivät ja pystyakselilla on jokaisen aktiivisen työntekijän nimi. Näiden tietojen risteävissä kohdissa on pudotusvalikot, joista voidaan valita jokaiselle työntekijälle ajettava linjavuoro, vapaapäivä tai palkallinen vapaapäivä. Kohta voidaan jättää myös tyhjäksi, mikäli on etukäteen tiedossa, että kyseinen työntekijä on esimerkiksi sairauslomalla, eikä näin ollen pääse tulemaan töihin. Sairauslomapäiviä ei ole huomioitu ohjelmassa, vaan niiden tarvittavista merkinnöistä vastaa palkanmaksaja. Aina, kun linjavuoro on kyseiselle päivälle valittuna jollekin työntekijälle, se poistuu pudotusvalikosta, eikä ole valittavissa toiselle työntekijälle. Näin käyttö on helpompaa, eikä päällekkäisyyksiä tapahdu.

Tallennettua työvuorolistaa pääsee muokkaamaan jälkikäteen, mutta ohjelma ei huomioi, eikä siihen tarvitse merkitä, mikäli työntekijät päättävät vaihtaa keskenään työvuoroja. Tämänkin asian tarvittavista merkinnöistä vastaa palkanmaksaja.

#### 4.4 Palkanmaksu

Palkanmaksu-osiossa tulee ensin valita työntekijä ja haluttu palkkakausi, eli mihin työvuorolistaan laskelmat perustuvat. Kun nämä on valittu, ohjelma suorittaa palkkalaskelman. Palkkalaskelmassa on ilmoitettu paljonko henkilön brutto- ja nettopalkaksi muodostuu. Palkkaerittelyssä voidaan tarkastella henkilön kokonaistuntimäärää, sekä tunteja, joita koskee yhteisliikennealuelisä, sunnuntai-, iltai-, yö-, tai ylityölisä. Lisäksi on lueteltu tehtävät vähennykset ja mahdollinen päiväraha. Näitä tietoja voidaan juoksevasti muokata, eli mikäli halutaan muuttaa jotakin henkilöltä tehtävän vähennyksen tai lisän määrää. Tehtäviä vähennyksiä ovat ennakonpidätys, eläkemaksu, työttömyysvakuutusmaksu ja mahdollinen liiton ay-maksu. Nähtävillä on myös paljonko henkilön veroprosentti, liisäveroprosentti ja tuloraja ovat. Ohjelma ei laske valmiiksi mahdollisen ulosottopidätyksen määrää, mutta ilmoittaa siitä palkanmaksulaskelmassa, jolloin palkanmaksaja suorittaa pidätyksen.

## 5 TUOTEKEHITYS

### 5.1 Käyttäjäystävällisyys

Tehtäessä sovellusta asiakkaalle, on myös käyttöliittymäsuunnitteluun kiinnitettävä huomiota. Ohjelman tulee olla mahdollisimman helppo käyttää ja vastattava asiakkaan tarpeisiin. Sovelluksen kehityksessä oltiin aktiivisessa vuorovaikutuksessa asiakkaan kanssa, jotta sovelluksesta tuli asiakkaan toiveiden mukainen ja hyväksymä. Toisen alan opiskelijoilla ei välttämättä ole tietämystä linja-autoalan työehtosopimuksista tai muista käytännön asioista, minkä vuoksi oli tärkeää, että asiakas itse kertoi, mitä ohjelman tulisi sisältää.

### 5.2 Muille yrityksille

Liikennöintiyrityksen omien tietojen, kuten linjavuorot sekä työntekijälistat, lisäksi tulee ottaa huomioon muita asioita, kun toiselle yritykselle räätälöityä tuotetta kehitetään muille liikennöintiyrityksille tai muun alan yrityksille.

Toiselle liikennöintialan yritykselle tuotetta kehittäessä, tulee ottaa huomioon miten työehtosopimuksessa mainitut seikat on sovellettu yrityksen toiminnassa. Työehtosopimus on luotu ensisijaisesti työntekijän oikeuksien perustaksi. Työehdot voivat kuitenkin olla muuta kuin työehtosopimuksessa on mainittu, kunhan ne ovat vähintään niitä vastaavat. Esimerkiksi työehtosopimuksessa on mainittu vähimmäispalkkaus, mutta palkkaus voi olla yrityksellä sitä korkeampi.

Rakenteellisia muutoksia voidaan joutua tekemään myös lähdekoodiin, mikäli kyseessä on toinen liikennöintialan yritys, joka toimii eri paikkakunnalla, jolloin siihen pätee muun muassa eri yhteisliikennöintialueiden tuntilisät.

Eri alan yritykselle ohjelmaa kehitettäessä lähdekoodia ja tietokantarakenteita jouduttaiisiin sopeuttamaan yrityksen tarpeita vastaaviksi.

### 5.3 Ylläpito ja muutokset

Kun ohjelmisto myydään yritykselle, voidaan tehdä erillinen huoltosopimus tai muu tae ohjelmiston ylläpidosta ja päivityksistä mahdollisten muutosten ilmetessä. Ohjelmisto voidaan myydä myös kertakauppana, jolloin myyjällä ei ole velvollisuuksia ohjelmiston toimivuuden varmistamiseksi tulevaisuudessa siirtymäajan jälkeen.

Työehtosopimus tyypillisesti muuttuu muutamien vuosien välein. Mikäli muutokset koskevat sovelluksessa käsiteltäviä seikkoja, tulee sovellusta muuttaa nykyistä noudatettavaa työehtosopimusta vastaavaksi. Yrityksen tulee itse huolehtia muutosten toimeenpanosta omassa palkanmaksussaan tai pyytää ne lisättäväksi ohjelmaan erillistä korvausta vastaan.

Työehtosopimuksen tapaan myös lainsäädäntö alaa koskien voi muuttua, jolloin muutokset on ehdottomasti otettava yrityksen palkanmaksussa ja työvuorolistoissa huomioon joko ohjelmaan lisäten tai toimihenkilöiden hoidettavaksi.

Yritys voi myös itse muuttaa omia käytäntöjään ja toimintatapojaan.

## 6 POHDINTA

Etenkin monen pienyrityksen palkanmaksuohjelmat ovat päivityksen tarpeessa, joten kyseisellä tuotteella löytyy selkeä tarve. Ohjelman tekeminen vaatii aikaa ja tiivistä kommunikointia sekä asiakkaan että muun kehitystiimin kanssa, jotta ohjelmasta saadaan juuri sopiva ja räätälöity versio kyseisen asiakkaan tarpeisiin.

Ohjelman tekemisessä ei osattu varautua käsiteltävän tiedon kokonaismäärään, jonka vuoksi ohjelman tekeminen viivästyi tarvittavien muutoksien aikaansaamiseksi. Lisäksi ohjelman palkanmaksuosio ei toiminut täysin vaatimusten mukaisesti vielä myöhemmissäkään versioissa, eikä ohjelma opinnäytetyön palauttamisen aikana ole vielä loppuasiakkaan käytössä. On kuitenkin arvioitu sen valmistuvan muutaman kuukauden kuluessa.

Valmista ohjelmaa voidaan jatkossa kehittää myös muille yrityksille, sekä tehdä siihen tarvittavia päivityksiä.

## LÄHTEET

AKT. 2014. Linja-autohenkilökunnan TES. Luettu 15.4.2016.  
[http://www.akt.fi/site/assets/files/1734/linja-autohenkilokunnan\\_tes\\_2014-2016.pdf](http://www.akt.fi/site/assets/files/1734/linja-autohenkilokunnan_tes_2014-2016.pdf)

Ari Hovi. 2015. MongoDB haastaa relaatiotietokantoja. Luettu 19.4.2016.  
<http://www.arihovi.com/mongodb-haastaa-relaatiokantoja/>

MongoDb. 2016. What is MongoDB. Luettu 19.4.2016.  
<https://www.mongodb.com/what-is-mongodb>

Wikipedia. 2016a. Node.js. Luettu 19.4.2016.  
<https://en.wikipedia.org/wiki/Node.js>

Wikipedia. 2016b. MongoDB. Luettu 19.4.2016.  
<https://en.wikipedia.org/wiki/MongoDB>

Wikipedia. 2016c. Meteor. Luettu 25.4.2016  
[https://en.wikipedia.org/wiki/Meteor\\_\(web\\_framework\)](https://en.wikipedia.org/wiki/Meteor_(web_framework))